



950 Rittenhouse Rd., Norristown, PA 19403 • Tel.: 215/666-7950 • TWX: 510/660-4168

6510 MICROPROCESSOR WITH I/O

DESCRIPTION

The 6510 is a low-cost microcomputer system capable of solving a broad range of small-systems and peripheral-control problems at minimum cost to the user.

An 8-bit Bi-Directional I/O Port is located on-chip with the Output Register at Address 0 0 0 0 and the Data-Direction Register at Address 0 0 0 1. The I/O Port is bit-by-bit programmable.

The Three-State sixteen-bit Address Bus allows Direct Memory Accessing (DMA) and multi-processor systems sharing a common memory.

The internal processor architecture is identical to the MOS Technology 6502 to provide software compatibility.

FEATURES OF THE 6510 . . .

- 8-Bit Bi-Directional I/O Port
- Single +5 volt supply
- N channel, silicon gate, depletion load technology
- Eight bit parallel processing
- 56 Instructions
- Decimal and binary arithmetic
- Thirteen addressing modes
- True indexing capability
- Programmable stack pointer
- Variable length stack
- Interrupt capability
- 8 Bit Bi-Directional Data Bus
- Addressable memory range of up to 65K bytes
- Direct memory access capability
- Bus compatible with M6800
- Pipeline architecture
- 1 MHz and 2MHz operation
- Use with any type or speed memory

PIN CONFIGURATION

RES	1	40	ϕ_2 IN
ϕ_1 IN	2	39	R/W
IRQ	3	38	DB ₈
AEC	4	37	DB ₇
VDD	5	36	DB ₆
A ₀	6	35	DB ₅
A ₁	7	34	DB ₄
A ₂	8	33	DB ₃
A ₃	9	32	DB ₂
A ₄	10	31	DB ₁
A ₅	11	30	P ₀
A ₆	12	29	P ₁
A ₇	13	28	P ₂
A ₈	14	27	P ₃
A ₉	15	26	P ₄
A ₁₀	16	25	P ₅
A ₁₁	17	24	P ₆
A ₁₂	18	23	P ₇
A ₁₃	19	22	A ₁₅
VSS	20	21	A ₁₄



6510 CHARACTERISTICS

MAXIMUM RATINGS

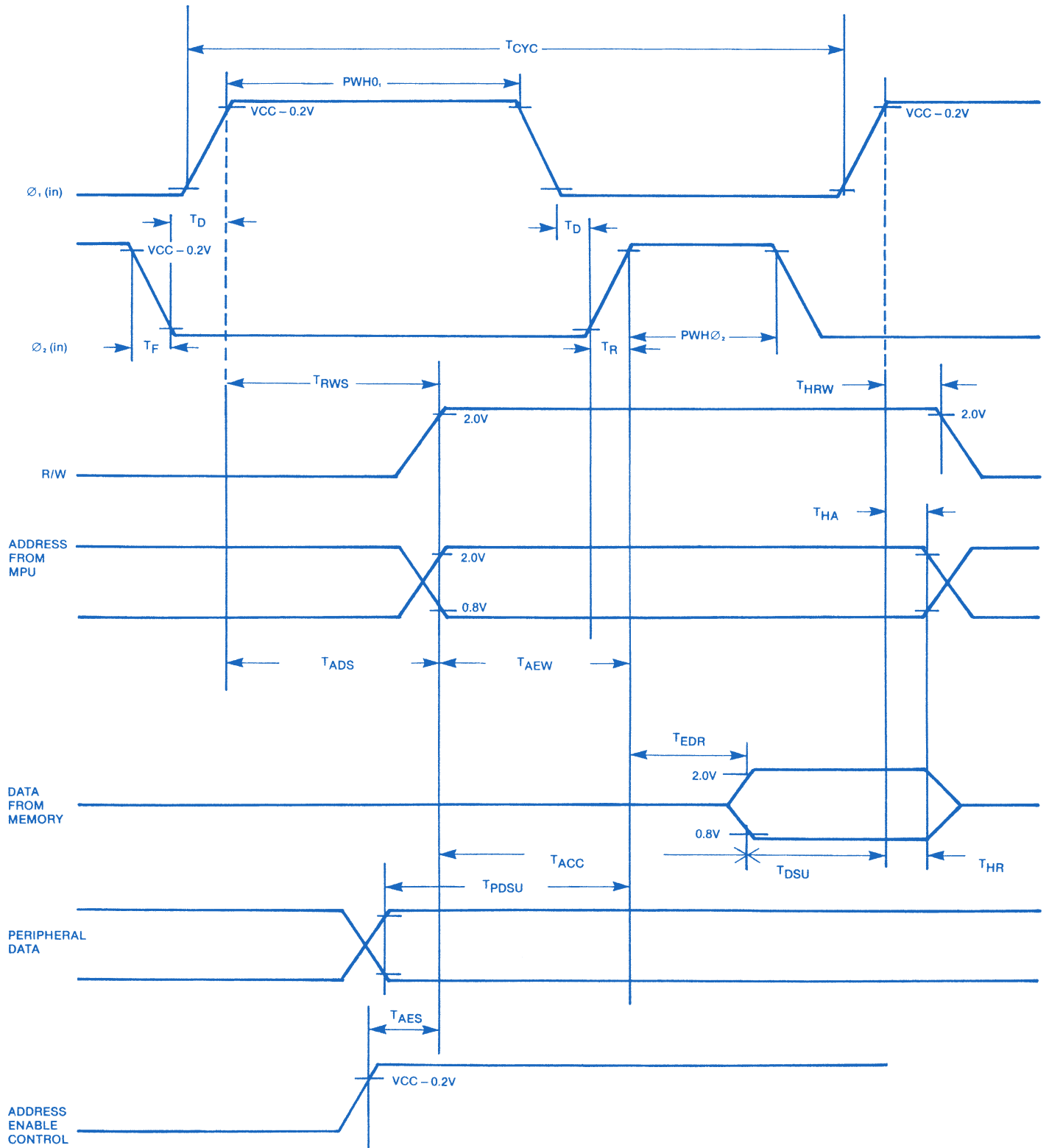
RATING	SYMBOL	VALUE	UNIT
SUPPLY VOLTAGE	V _{CC}	− 0.3 to + 7.0	Vdc
INPUT VOLTAGE	V _{IN}	− 0.3 to + 7.0	Vdc
OPERATING TEMPERATURE	T _A	0 to + 70	°C
STORAGE TEMPERATURE	T _{STG}	− 55 to + 150	°C

This device contains input protection against damage due to high static voltages or electric fields; however, precautions should be taken to avoid application of voltages higher than the maximum rating.

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0V ± 5%, V_{SS} = 0, T_A = 0° to + 70°C)

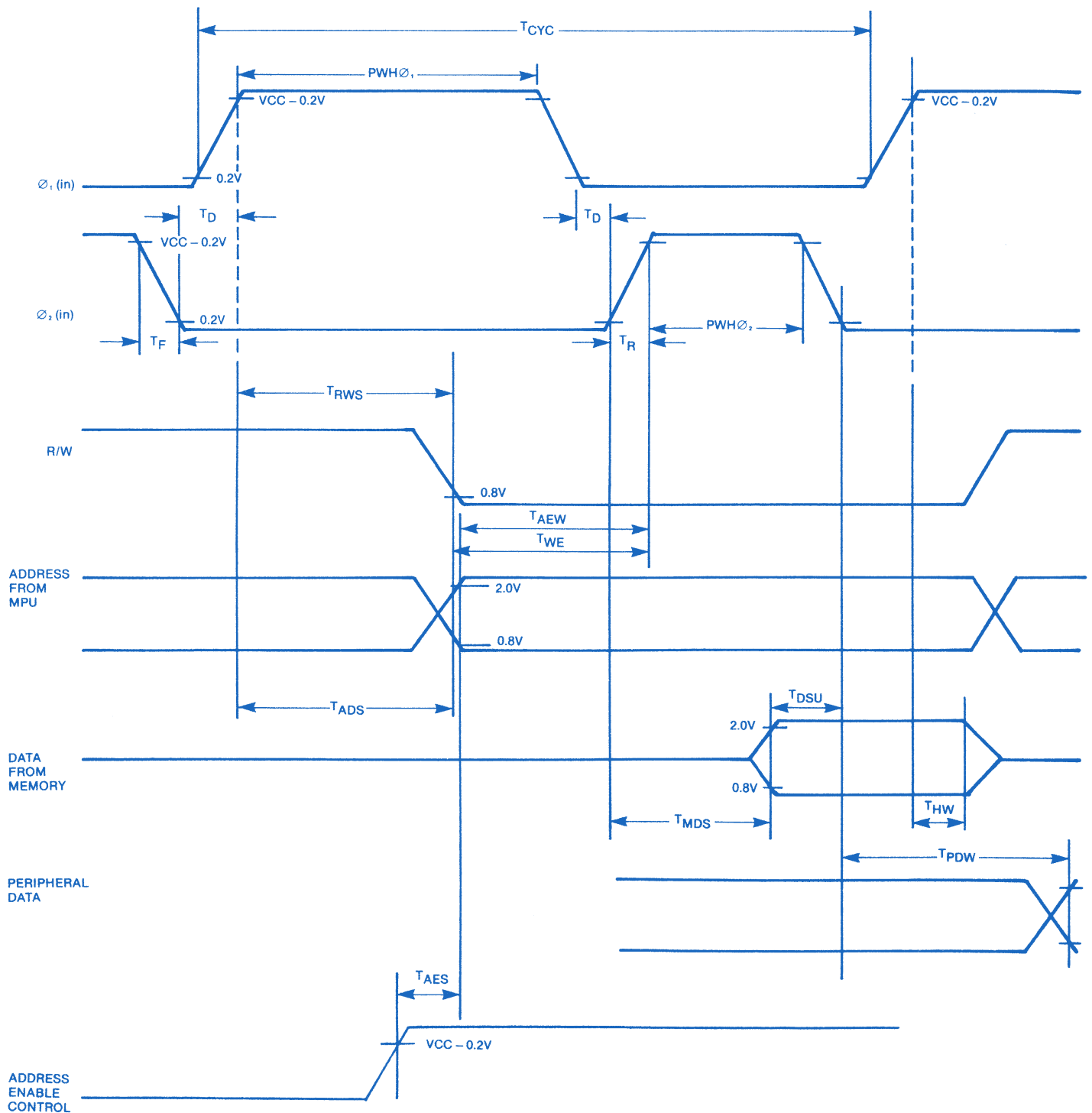
CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT
Input High Voltage $\phi_1, \phi_2(\text{in})$	V _{IH}	V _{CC} − 0.2	—	V _{CC} + 1.0V	Vdc
Input High Voltage $\overline{\text{RES}}, \text{P}_0\text{-P}, \overline{\text{IRQ}}, \text{Data}$		V _{SS} + 2.0	—	—	Vdc
Input Low Voltage $\phi_1, \phi_2(\text{in})$	V _{IL}	V _{SS} − 0.3	—	V _{SS} + 0.2	Vdc
$\overline{\text{RES}}, \text{P}_0\text{-P}, \overline{\text{IRQ}}, \text{Data}$		—	—	V _{SS} + 0.8	Vdc
Input Leakage Current (V _{IN} = 0 to 5.25V, V _{CC} = 5.25V) Logic $\phi_1, \phi_2(\text{in})$	I _{IN}	— —	— —	2.5 100	μA μA
Three State (Off State) Input Current (V _{IN} = 0.4 to 2.4V, V _{CC} = 5.25V) Data Lines	ITSI	—	—	10	μA
Output High Voltage (I _{OH} = −100 μA dc, V _{CC} = 4.75V) Data, AO-A15, R/W, P ₀ -P ₇	VOH	V _{SS} + 2.4	—	—	Vdc
Out Low Voltage (I _{OL} = 1.6mA _{dc} , V _{CC} = 4.75V) Data, AO-A15, R/W, P ₀ -P ₇	VOL	—	—	V _{SS} + 0.4	Vdc
Power Supply Current	ICC	—	125		mA
Capacitance V _{IN} = 0, T _A = 25°C, f = 1MHz Logic, P ₀ -P ₇	C C _{IN}	— —	— —	10	pF
Data AO-A15, R/W	C _{OUT}	— —	— —	15 12	
ϕ_1	C ϕ_1	—	30	50	
ϕ_2	C ϕ_2	—	50	80	

CLOCK TIMING



**TIMING FOR READING DATA FROM
MEMORY OR PERIPHERALS**

CLOCK TIMING



**TIMING FOR WRITING DATA TO
MEMORY OR PERIPHERALS**

AC CHARACTERISTICS

1 MHz TIMING

2 MHz TIMING

ELECTRICAL CHARACTERISTICS (VCC = 5V ± 5%, VSS = 0V, T_A = 0° – 70°C)

CLOCK TIMING

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	UNITS
Cycle Time	T _{CYC}	1000	—	—	500	—	—	ns
Clock Pulse Width (Measured at VCC – 0.2V)	PWH _{Ø1} PWH _{Ø2}	430 470	— —	— —	215 235	— —	— —	ns ns
Fall Time, Rise Time (Measured from 0.2V to VCC – 0.2V)	T _F , T _R	—	—	25	—	—	15	ns
Delay Time between Clocks (Measured at 0.2V)	T _D	0	—	—	0	—	—	ns

READ/WRITE TIMING (LOAD = 1TTL)

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	UNITS
Read/Write Setup Time from 6508	T _{RWS}	—	100	300	—	100	150	ns
Address Setup Time from 6508	T _{ADS}	—	100	300	—	100	150	ns
Memory Read Access Time	T _{ACC}	—	—	575	—	—	300	ns
Data Stability Time Period	T _{DSU}	100	—	—	50	—	—	ns
Data Hold Time-Read	T _{HR}	—	—	—	—	—	—	ns
Data Hold Time-Write	T _{HW}	10	30	—	10	30	—	ns
Data Setup Time from 6510	T _{MDS}	—	150	200	—	75	100	ns
Address Hold Time	T _{HA}	10	30	—	10	30	—	ns
R/W Hold Time	T _{HRW}	10	30	—	10	30	—	ns
Delay Time, Address valid to Ø2 positive transition	T _{AEW}	180	—	—	—	—	—	ns
Delay Time, Ø2 positive transition to Data valid on bus	T _{EDR}	—	—	395	—	—	—	ns
Delay Time, Data valid to Ø2 negative transition	T _{DSU}	300	—	—	—	—	—	ns
Delay Time, R/W negative transition to Ø2 positive transition	T _{WE}	130	—	—	—	—	—	ns
Delay Time, Ø2 negative transition to Peripheral Data valid	T _{PDW}	—	—	1	—	—	—	µs
Peripheral Data Setup Time	T _{PDSU}	300	—	—	—	—	—	ns
Address Enable Setup Time	T _{AES}	—	—	60	—	—	60	ns

SIGNAL DESCRIPTION

Clocks (ϕ_1, ϕ_2)

The 6510 requires a two phase non-overlapping clock that runs at the Vcc voltage level.

Address Bus (A_0 - A_{15})

These outputs are TTL compatible, capable of driving one standard TTL load and 130 pf.

Data Bus (D_0 - D_7)

Eight pins are used for the data bus. This is a Bi-Directional bus, transferring data to and from the device and peripherals. The outputs are tri-state buffers capable of driving one standard TTL load and 130 pf.

Reset

This input is used to reset or start the microprocessor from a power down condition. During the time that this line is held low, writing to or from the microprocessor is inhibited. When a positive edge is detected on the input, the microprocessor will immediately begin the reset sequence.

After a system initialization time of six clock cycles, the mask interrupt flag will be set and the microprocessor will load the program counter from the memory vector locations FFFC and FFFD. This is the start location for program control.

After Vcc reaches 4.75 volts in a power up routine, reset must be held low for at least two clock cycles. At this time the R/W signal will become valid.

When the reset signal goes high following these two clock cycles, the microprocessor will proceed with the normal reset procedure detailed above.

Interrupt Request (\overline{IRQ})

This TTL level input requests that an interrupt sequence begin within the microprocessor. The microprocessor will complete the current instruction being executed before recognizing the request. At that time, the interrupt mask bit in the Status Code Register will be examined. If the interrupt mask flag is not set, the microprocessor will begin an interrupt sequence. The Program Counter and Processor Status Register are stored in the stack. The microprocessor will then set the interrupt mask flag high so that no further interrupts may occur. At the end of this cycle, the program counter low will be loaded from address FFFE, and program counter high from location FFFF, therefore transferring program control to the memory vector located at these addresses.

Address Enable Control (AEC)

The Address Bus is valid only when the Address Enable Control line is high. When low, the Address Bus is in a high-impedance state. This feature allows easy DMA and multiprocessor systems.

I/O Port (P_0 - P_7)

Eight pins are used for the peripheral port, which can transfer data to or from peripheral devices. The Output Register is located in RAM at Address 0001, and the Data Direction Register is at Address 0000. The outputs are capable at driving one standard TTL load and 130 pf.

Read/Write (R/W)

This signal is generated by the microprocessor to control the direction of data transfers on the Data Bus. This line is high except when the microprocessor is writing to memory or a peripheral device.

ADDRESSING MODES

ACCUMULATOR ADDRESSING—This form of addressing is represented with a one byte instruction, implying an operation on the accumulator.

IMMEDIATE ADDRESSING—In immediate addressing, the operand is contained in the second byte of the instruction, with no further memory addressing required.

ABSOLUTE ADDRESSING—In absolute addressing, the second byte of the instruction specifies the eight low order bits of the effective address while the third byte specifies the eight high order bits. Thus, the absolute addressing mode allows access to the entire 65K bytes of addressable memory.

ZERO PAGE ADDRESSING—The zero page instructions allow for shorter code and execution times by only fetching the second byte of the instruction and assuming a zero high address byte. Careful use of the zero page can result in significant increase in code efficiency.

INDEXED ZERO PAGE ADDRESSING—(X, Y indexing)—This form of addressing is used in conjunction with the index register and is referred to as "Zero Page, X" or "Zero Page, Y." The effective address is calculated by adding the second byte to the contents of the index register. Since this is a form of "Zero Page" addressing, the content of the second byte references a location in page zero. Additionally, due to the "Zero Page" addressing nature of this mode, no carry is added to the high order 8 bits of memory and crossing of page boundaries does not occur.

INDEXED ABSOLUTE ADDRESSING—(X, Y indexing)—This form of addressing is used in conjunction with X and Y index register and is referred to as "Absolute, X," and "Absolute, Y." The effective address is formed by adding the contents of X and Y to the address contained in the second and third bytes of the instruction. This mode allows the index register to contain the index or count value and the instruction to contain the base address. This type of indexing allows any location referencing and the index to modify multiple fields resulting in reduced coding and execution time.

IMPLIED ADDRESSING—In the implied addressing mode, the address containing the operand is implicitly stated in the operation code of the instruction.

RELATIVE ADDRESSING—Relative addressing is used only with branch instructions and establishes a destination for the conditional branch.

The second byte of the instruction becomes the operand which is an "Offset" added to the contents of the lower eight bits of the program counter when the counter is set at the next instruction. The range of the offset is - 128 to + 127 bytes from the next instruction.

INDEXED INDIRECT ADDRESSING—In indexed indirect addressing (referred to as [Indirect, X]), the second byte of the instruction is added to the contents of the X index register, discarding the carry. The result of this addition points to a memory location on page zero whose contents is the low order eight bits of the effective address. The next memory location in page zero contains the high order eight bits of the effective address. Both memory locations specifying the high and low order bytes of the effective address must be in page zero.

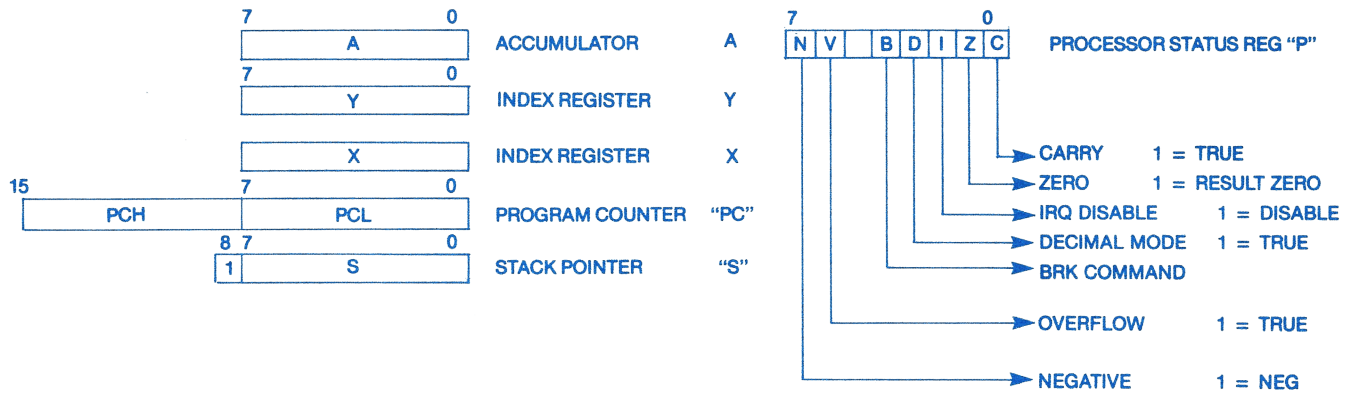
INDIRECT INDEXED ADDRESSING—In indirect indexed addressing (referred to as [Indirect, Y]), the second byte of the instruction points to a memory location in page zero. The contents of this memory location is added to the contents of the Y index register, the result being the low order eight bits of the effective address. The carry from this addition is added to the contents of the next page zero memory location, the result being the high order eight bits of the effective address.

ABSOLUTE INDIRECT—The second byte of the instruction contains the low order eight bits of a memory location. The high order eight bits of that memory location is contained in the third byte of the instruction. The contents of the fully specified memory location is the low order byte of the effective address. The next memory location contains the high order byte of the effective address which is loaded into the sixteen bits of the program counter.

INSTRUCTION SET—ALPHABETIC SEQUENCE

ADC	Add Memory to Accumulator with Carry	LDA	Load Accumulator with Memory
AND	"AND" Memory with Accumulator	LDX	Load Index X with Memory
ASL	Shift left One Bit (Memory or Accumulator)	LDY	Load Index Y with Memory
BCC	Branch on Carry Clear	LSR	Shift One Bit Right (Memory or Accumulator)
BCS	Branch on Carry Set	NOP	No Operation
BEQ	Branch on Result Zero	ORA	"OR" Memory with Accumulator
BIT	Test Bits in Memory with Accumulator	PHA	Push Accumulator on Stack
BMI	Branch on Result Minus	PHP	Push Processor Status on Stack
BNE	Branch on Result not Zero	PLA	Pull Accumulator from Stack
BPL	Branch on Result Plus	PLP	Pull Processor Status from Stack
BRK	Force Break	ROL	Rotate One Bit Left (Memory or Accumulator)
BVC	Branch on Overflow Clear	ROR	Rotate One Bit Right (Memory or Accumulator)
BVS	Branch on Overflow Set	RTI	Return from Interrupt
CLC	Clear Carry Flag	RTS	Return from Subroutine
CLD	Clear Decimal Mode	SBC	Subtract Memory from Accumulator with Borrow
CLI	Clear Interrupt Disable Bit	SEC	Set Carry Flag
CLV	Clear Overflow Flag	SED	Set Decimal Mode
CMP	Compare Memory and Accumulator	SEI	Set Interrupt Disable Status
CPX	Compare Memory and Index X	STA	Store Accumulator in Memory
CPY	Compare Memory and Index Y	STX	Store Index X in Memory
DEC	Decrement Memory by One	STY	Store Index Y in Memory
DEX	Decrement Index X by One	TAX	Transfer Accumulator to Index X
DEY	Decrement Index Y by One	TAY	Transfer Accumulator to Index Y
EOR	"Exclusive-or" Memory with Accumulator	TSX	Transfer Stack Pointer to Index X
INC	Increment Memory by One	TXA	Transfer Index X to Accumulator
INX	Increment Index X by One	TXS	Transfer Index X to Stack Register
INY	Increment Index Y by One	TYA	Transfer Index Y to Accumulator
JMP	Jump to New Location		
JSR	Jump to New Location Saving Return Address		

PROGRAMMING MODEL



INSTRUCTION SET – OP CODES, Execution Time, Memory Requirements

INSTRUCTIONS		IMMEDIATE		ABSOLUTE		ZERO PAGE		ACCUM.		IMPLIED		(IND.X)		(IND.Y)		3-PAGE.X		ABS.X		ABS.Y		RELATIVE		INDIRECT		3-PAGE.Y		CONDITION CODES						
MNEMONIC	OPERATION	OP	N	8	OP	N	8	OP	N	8	OP	N	8	OP	N	8	OP	N	8	OP	N	8	OP	N	8	OP	N	8	N	Z	C	I	D	V
A D C	A ← M + C → A	(1)	69	2	2	0D	4	3	05	3	2					61	6	2	71	5	2	75	4	2	7D	4	3	79	4	3				
A N D	A ← A & M	(1)	29	2	2	0D	4	3	26	3	2					21	6	2	31	5	2	35	4	2	3D	4	3	39	4	3				
A S L	C ← (7) → 0																																	
B B C	BRANCH ON C=0	(2)																																
B B C	BRANCH ON C=1	(2)																																
B B O	BRANCH ON Z=1	(2)																																
B B T	A ← M																																	
B M I	BRANCH ON N=1	(2)																																
B N E	BRANCH ON Z=0	(2)																																
B P L	BRANCH ON N=0	(2)																																
B R K	(See Fig. 1)																																	
B V C	BRANCH ON V=0	(2)																																
B V S	BRANCH ON V=1	(2)																																
C L C	0 ← C																																	
C L D	0 ← D																																	
C L I	0 ← I																																	
C L V	0 ← V																																	
C M P	A ← M	(1)	C9	2	2	CD	4	3	C5	3	2					C1	6	2	D1	5	2	D5	4	2	DD	4	3	D9	4	3				
C P X	X ← M		E9	2	2	EC	4	3	E4	3	2																							
C P Y	Y ← M		C9	2	2	CC	4	3	C4	3	2																							
D E C	M ← 1 ← M																																	
D E X	X ← 1 ← X																																	
D E Y	Y ← 1 ← Y																																	
E O R	A ← M ⊕ A	(1)	A9	2	2	AD	4	3	A5	3	2					41	6	2	61	5	2	65	4	2	6D	4	3	69	4	3				
I N C	M ← 1 ← M																																	
I N X	X ← 1 ← X																																	
I N Y	Y ← 1 ← Y																																	
J M P	JUMP TO NEW LOC.																																	
J S R	(See Fig. 2) JUMP SUB																																	
L D A	M ← A	(1)	A9	2	2	AD	4	3	A5	3	2					A1	6	2	B1	5	2	B5	4	2	BD	4	3	B9	4	3				
L D X	M ← X	(1)	A2	2	2	AE	4	3	A6	3	2																							
L D Y	M ← Y	(1)	A9	2	2	AD	4	3	A5	3	2																							
L S R	0 ← (7) ← C																																	
N O P	NO OPERATION																																	
O R A	A ← M		99	2	2	9D	4	3	95	3	2					01	6	2	11	5	2	15	4	2	1D	4	3	19	4	3				
P H A	A ← M _S S ← 1 ← S																																	
P H Y	Y ← M _S S ← 1 ← S																																	
P L A	S ← 1 ← S M _S ← A																																	
P L Y	S ← 1 ← S M _S ← Y																																	
R O L	(7) ← (7) ← C																																	
R O R	(7) ← (7) ← C																																	
R T I	(See Fig. 1) RTRN INT																																	
R T S	(See Fig. 2) RTRN SUB																																	
S B C	A ← M _S C ← A	(1)	99	2	2	9D	4	3	95	3	2					01	6	2	11	5	2	15	4	2	1D	4	3	19	4	3				
S E C	1 ← C																																	
S E D	1 ← D																																	
S E I	1 ← I																																	
S T A	A ← M																																	
S T X	X ← M																																	
S T Y	Y ← M																																	
T A X	A ← X																																	
T A Y	A ← Y																																	
T S X	S ← X																																	
T X A	X ← A																																	
T X S	X ← S																																	
T Y A	Y ← A																																	

(1) ADD 1 TO "N" IF PAGE BOUNDARY IS CROSSED

(2) ADD 1 TO "N" IF BRANCH OCCURS TO SAME PAGE.
ADD 2 TO "N" IF BRANCH OCCURS TO DIFFERENT PAGE.

(3) CARRY NOT=BORROW

(4) IF IN DECIMAL MODE Z FLAG IS INVALID.
ACCUMULATOR MUST BE CHECKED FOR ZERO RESULT.

X INDEX, X

Y INDEX, Y

A ACCUMULATOR

M MEMORY PER EFFECTIVE ADDRESS

M_S MEMORY PER STACK POINTER

+ ADD

- SUBTRACT

A AND

V OR

↕ EXCLUSIVE OR

✓ MODIFIED

- NOT MODIFIED

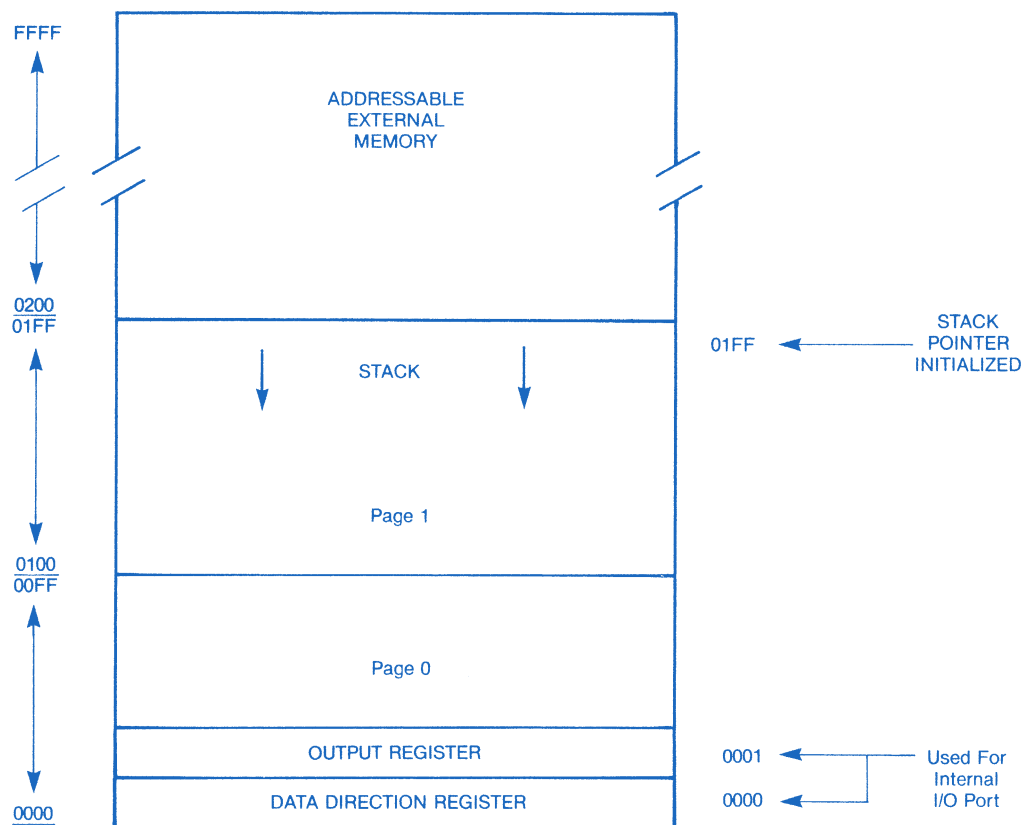
M₇ MEMORY BIT 7

M₆ MEMORY BIT 6

N NO CYCLES

NO. BYTES

Note: Commodore Semiconductor Group cannot assume liability for the use of undefined OP Codes



6510 MEMORY MAP

APPLICATIONS NOTES

Locating the Output Register at the internal I/O Port in Page Zero enhances the powerful Zero Page Addressing instructions of the 6510.

By assigning the I/O Pins as inputs (using the Data Direction Register) the user has the ability to change the contents of address 0001 (the Output Register) using peripheral devices. The ability to change these contents using peripheral inputs, together with Zero Page Indirect Addressing instructions, allows novel and versatile programming techniques not possible earlier.

COMMODORE SEMICONDUCTOR GROUP reserves the right to make changes to any products herein to improve reliability, function or design. COMMODORE SEMICONDUCTOR GROUP does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.



A COMMODORE COMPANY

USER'S NOTES

brought to you by

<http://commodore.international/>

commodore international historical society

**this document was generously
contributed by
Michael Tomczyk**